UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA

"SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO LINEAR EM COMPUTADOR ANALÓGICO"

Tese submetida a Universidade Federal de Santa Catarina para a obtenção do grau de Mestre em Engenharia Industrial - Opção Pesquisa Operacional.

ICLÉA MARIA TERRA DE OLIVEIRA

FLORIANÓPOLIS
SANTA CATARINA - BRASIL
Abril - 1974

SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO LINEAR EM COMPUTADOR ANALÓGICO

ICLÉA MARIA TERRA DE OLIVEIRA

ESTA TESE FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

"MESTRE EM ENGENHARIA"

ESPECIALIDADE ENGENHARIA INDUSTRIAL - OPÇÃO PESQUISA OPERACIONAL APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO

Prof. WALTER CELSO DE LIMA, M.Sc.
Orientador

Prof. DOMINGOS BOECHAT ALVES, Ph.D.
Integrador dos Cursos de Pós-Graduação

BANCA EXAMINADORA

Prof. LUIZ GONZAGA DE SOUZA FONSECA, M.Sc.

Prof. RAJAMANI DORAISWAMI, Ph.D.

Prof. RAUL VALENTIM DA SILVA, M.Sc.

AGRADECIMENTOS

À CAPES e ao BNDE pelo apoio financeiro.

À Escola Federal de Engenharia de Itajubá, Minas Gerais, pela utilização de seu computador.

A todos que contribuiram para a realização deste trabalho.

RESUMO

Neste trabalho, faz-se um estudo comparativo entre os métodos de solução de problemas de programação linear em computadores analógicos e digitais.

A primeira parte apresenta os fundamentos teóricos da resolução de problemas de programação linear, bem como o método de solução digital proposto por Dantzig - método "simplex". Utilizou-se como exemplo o programa IBM - LPMOSS.

A segunda parte consta da apresentação e discussão do método de solução analógica de problemas de programação convexa, particularizado para programação linear, bem como, dos circuitos eletrônicos utilizados.

Os métodos são aplicados a problemas-exemplo. Da solução destes, são tiradas algumas conclusões práticas e teóricas que apontam as vantagens e desvantagens de cada um dos métodos.

Os sistemas computacionais utilizados foram:

- Sistema Digital: IBM 1130
- Sistema Analógico: Telefunken RA 770.

ABSTRACT

This works presents a comparative study on methods of solving linear programming problems on analog and digital computers.

The first part presents the theoretical basis to obtain a solution of a linear programming problem. The "simplex" method proposed by Dantzig and the IBM-LPMOSS program are used to show a digital solution method.

The second part deals with the theoretical of the method of solving convex programming problems on analog computers. The method is applied to a linear programming problem and some circuits for analog computers are suggested.

Both methods are applied on solving linear programming problems. And their solutions give an idea of the advantages of each method, by means of their precision and time of solution.

The IBM 1130 computer was used to obtain the digital solution and the Telefunken RA 770 to the analog one.

SUMÁRIO

CAPÍTULO I

INTRODUÇÃO

Um sistema, quer seja econômico, biológico ou físico, é caracterizado por uma série de atividades inter-relacionadas. Estas atividades, quando possível, são identificadas pelas variáveis do modelo matemático. É sempre esperado o melhor desempenho do sistema. Em geral, são apresentadas várias soluções alternativas; é necessário, pois, desenvolver um método para escolher a melhor das alternativas apontadas segundo um determinado critério.

Os problemas de programação matemática tratam da escolha de uma solução ótima dentre aquelas que satisfazem às condições do problema.

Entre os problemas de programação matemática, encontram-se os problemas de programação convexa que englobam problemas de programação linear, quadrática e outros.

É desnecessário enfatizar a importância da solução destes problemas; inúmeros casos reais podem ser encontrados em bibliografia especializada[4,7].

O desenvolvimento dos computadores digitais coincide com a dedução dos primeiros métodos de solução dos problemas de programação linear de forma digital, ou melhor dizendo, de forma numérica.

Dentre os métodos, quase todos eles desenvolvidos em função do computador digital, aquele que mais se difundiu foi o método conhecido como "simplex", apresentado por G. DANTZIG[4] em 1947. Este método foi revisado e é largamente utilizado até hoje, tendo sido desenvolvido "Software" para a solução do problema de programação linear[13].

O desenvolvimento do computador analógico permitiu que se estudasse a possibilidade de resolver os problemas de programação matemática neste tipo de computador.

Em 1955, C. ABLOW e G. BRIGHAM[1] apresentam o problema genericamente, apenas sugerindo um método de solução sem justificá-lo teoricamente. Em 1956, I. PYNE[19] particulariza o problema para um método de programação linear e o justifica geometricamente. Em 1964, D. LONGLEY[17] resolve um problema de transporte no computador analógico e compara os resultados obtidos com aqueles do computador digital.

A partir de 1960, começam a ser publicados na URSS uma série de artigos de M. V. RYBASHOV[21,25], I. KARPINSKAYA[14,15] e E. DUDNIKOV[5] que visam a apresentar algumas possibilidades de solução do problema de programação convexa, em particular, programação linear e quadrática em computador analógico.

Recentemente, com o desenvolvimento de computadores híbridos, foram publicados trabalhos[16,9,8] que, utilizando as vantagens das unidades analógicas e digitais, resolvem o problema de programação convexa, usando como base as mesmas idéias de Rybashov.

Outros trabalhos[18,27] exploram o fundamento teórico do método analógico de solução de problemas de programação linear, visando a resolver aqueles problemas que não podem ser solucionados pelo método de Dantzig.

Propõe-se, neste trabalho, desenvolver os fundamentos teóricos do método de Rybashov[23] com base na teoria geral de programação matemática. Solucionam-se alguns problemas de programação linear, como uma particularização do método, em um computador analógico-híbrido Telefunken RA 770.

A primeira parte deste trabalho contém a apresentação do método "simplex", sua aplicação e a demonstração da impossibilidade de resolver certos problemas. A segunda parte é dedicada à solução analógica do problema. Inicia-se pela justificativa da não-aplicabilidade do método "simplex" ao computador analógico, tendo em vista a instabilidade do sistema. É encontrada uma forma de resolver o problema com a estabilidade assegurada. Este método é apresentado no capítulo seguinte onde também é discutida a utilização de determinados componentes eletrônicos. A seguir, são resolvidos problemas que permitem chegar-se a algumas conclusões as quais são apresentadas no último capítulo.

PARTE I

CAPÍTULO II

PROGRAMAÇÃO LINEAR

2.1. Definição do Problema

O objetivo da programação linear é otimizar - maximizar ou minimizar - uma função linear sujeita a restrições lineares, isto é, deseja-se otimizar uma função do tipo

$$Z = \sum_{j=1}^{n} c_j x_j \qquad 2.1$$

sujeita às restrições

$$\sum_{j=1}^{n} a_{ij} x_j + b_i \leq 0 \qquad (i = 1, \ldots\ldots m) \qquad 2.2$$

e $$x_j \geq 0 \qquad (j = 1, \ldots\ldots n) \qquad 2.3$$

onde a_{ij}, c_j e b_i são números reais.

A função linear 2.1 é chamada função objetivo; as restrições lineares 2.2 tanto podem ser equações como desigualdades. As variáveis x_j são sempre não-negativas (positivas ou nulas) de acordo com a restrição 2.3.

É conveniente fazer algumas observações quanto ao problema acima definido[12].

. Maximizar a função $Z = \sum_{j=1}^{n} c_j x_j$ é o mesmo que minimizar a função -Z, tomando o sinal contrário tal que:

min Z = max -Z.

. As restrições do tipo desigualdade podem ser transformadas em restrições do tipo igualdade através da introdução de variáveis de folga

$x_{n+1} (x_{n+1} \geq 0)$ da seguinte maneira:

$\sum_{j=1}^{n} a_{ij}x_j - x_{n+1} + b_i = 0$ é equivalente à $\sum_{j=1}^{n} a_{ij}x_j + b_i \geq 0$

e

$\sum_{j=1}^{n} a_{ij}x_j + x_{n+1} + b_i = 0$ é equivalente à $\sum_{j=1}^{n} a_{ij}x_j + b_i \leq 0$

. Uma desigualdade da forma $\sum_{j=1}^{n} a_{ij}x_j + b_i = 0$ é equivalente a duas desigualdades simultâneas

$$\sum_{j=1}^{n} a_{ij}x_j + b_i \geq 0$$

$$\sum_{j=1}^{n} a_{ij}x_j + b_i \leq 0$$

. k restrições simultâneas de forma $\sum_{j=1}^{n} a_{ij}x_j + b_i = 0$ (i= =1, ... k) são equivalentes a k +1 desigualdades simultâneas da forma

$$\sum_{j=1}^{n} a_{ij}x_j + b_i \geq 0 \qquad (i = 1, \ldots.k)$$

$$\sum_{j=1} \left(\sum_{j=1}^{n} a_{ij}x_j + b_i \right) \leq 0$$

. Se a variável x_j não é restrita quanto ao sinal, ela pode ser substituída por

$$x_j = x_j^{+} - x_j^{-}$$

onde

$$x_j^+ \geq 0 \quad e \quad x_j^- \geq 0$$

Na forma matricial, o problema de programação linear é escrito como

$$\text{Otimizar} \quad Z = \underline{c}^T\underline{X} \qquad 2.4$$

sujeito às restrições

$$\underline{A}\ \underline{X} + \underline{b} \leq \underline{0}$$

$$\underline{X} > \underline{0}$$

onde

$\underline{A}$ é uma matriz m x n;

$\underline{b}$ é um vetor coluna com m linhas;

$\underline{X}$ é um vetor coluna com n linhas;

$\underline{c}$ é um vetor coluna com n linhas.

2.2 - Teoremas Fundamentais

Considere-se o sistema de equações lineares na forma

$$\underline{A}^* \ \underline{X}^* + \underline{b}^* = \underline{0} \qquad 2.5$$

onde $\underline{A}^*$, $\underline{X}^*$ e $\underline{b}^*$ representam as matrizes e vetores que se obtêm acrescentando as variáveis de folga, com a finalidade de tornar igualdades todas as desigualdades. O sistema é dito consistente se tiver pelo menos uma solução e, redundante, se pelo menos uma das equações puder ser expressa como combinação linear de quaisquer outras.

Supondo que o sistema 2.5 é consistente e não redundante, seja $\underline{A}^* = \left[\underline{B}\ \underline{N}\right]$

onde $\underline{B}$ é uma matriz quadrada de ordem m e $\underline{N}$ é uma matriz de (n^*-m) colunas na qual n^* é o número de variáveis do problema mais as variáveis de folga

$$\underline{X} = \left[\underline{X}_B\ \underline{X}_N\right]$$

em que $\underline{X}_B$ é um vetor de m componentes e $\underline{X}_N$ é de n*-m componentes.

Chama-se solução básica do sistema 2.5 aquela que satisfaz às relações

$$\underline{X}_B + \underline{B}^{-1}\underline{b}^* = \underline{0}$$

$$\underline{X}_N = \underline{0}$$

As componentes de $\underline{X}_B$ são denominadas variáveis básicas e as de $\underline{X}_N$, variáveis não-básicas. A solução básica será chamada degenerada se alguma das componentes de $\underline{X}_B$ for zero. Os vetores cona de $\underline{B}$ são denominados vetores básicos e formam uma base do conjunto de soluções.

Embora nos sistema de desigualdades lineares possam ser introduzidas variáveis de folga para transformá-las em igualdades,eles podem ser tratados diretamente. A desigualdade linear da forma

$$\underline{a}_i\underline{X} + b_i \leq 0$$

define um semi-espaço fechado em relação aos hiperplanos

$$\underline{a}_i\underline{X} + b_i = 0$$

Se o sistema é consistente, as desigualdades definem um conjunto convexo fechado, e uma solução básica do sistema é um ponto extremo do conjunto onde n (número de variáveis do problema linear) são satisfeitas como igualdade. Se mais de n dos hoperplanos se interceptam em um ponto, tem-se uma solução básica degenerada[12]. Em programação linear, a solução de um sistema é dita possível se todas as componentes são não-negativas.

Teorema 2.1

Se existe uma solução de um sistema de equações lineares, existe uma solução básica deste sistema.

Teorema 2.2

Se existe uma solução possível de um sistema de equações lineares, existe uma solução básica possível.

Teorema 2.3

O conjunto de todas as soluções possíveis de um sistema de equações lineares constitui um conjunto convexo.

Teorema 2.4

Dado o poliedro convexo P de dimensão n definido pelo sistema de desigualdades

$$\underline{a}_i\underline{X} + b_i \leq 0 \qquad (i=1,\ldots m)$$

($\underline{a}_i$ é o i-ésimo vetor linha pertencente a $\underline{A}$), a condição necessária e suficiente para que o ponto $\underline{X} \in P$ seja um ponto extremo de P é que $\underline{X}$ seja solução de

$$\underline{a}_i\underline{X} + b_i = 0 \qquad i \in P$$

$$\underline{a}_i\underline{X} + b_i \leq 0 \qquad i \in P$$

$P = \{1,\ldots n;\ n < m\}$.

Teorema 2.5

Se todas as soluções possíveis de um problema de programação linear são limitadas, qualquer solução possível é uma combinação linear convexa de soluções básicas possíveis.

Teorema 2.6

Se existe uma solução ótima $\underline{X}^o$ para o problema de programação linear, então existe uma solução básica ótima.

As provas dos teoremas 2.1 a 2.6 se encontram facilmente na literatura especializada, particularmente em "Integer programming and network flows" de T.Hu[12].

2.3 - Método "Simplex"

O método "simplex" é um processo iterativo que consiste em passar de uma solução básica para outra adjacente, alterando o valor da função objetivo na tentativa de otimização.

O método "simplex" começa por uma solução básica possível que pode ser aquela de admitir como variáveis básicas as variáveis de folga adicionadas e, como não-básicas, as restantes, fazendo-as iguais a zero. Ou seja, a solução inicial deve satisfazer à equação matricial

$$\underline{I}\ \underline{X}_B + \underline{B}^{-1}\ \underline{N}\ \underline{X}_o + \underline{B}^{-1}\ \underline{b} = \underline{0}$$

onde $\underline{B}$ é uma matriz representativa da base e a função objetivo é expressa em função das variáveis não-básicas $\underline{X}_N$. No espaço das variáveis não-básicas, existe uma desigualdade da forma

$$\underline{B}^{-1}\ \underline{N}\ \underline{X}_N + \underline{B}^{-1}\ \underline{b} \leq \underline{0}$$

em que $\underline{X}_N = \underline{0}$ corresponde à origem do espaço de dimensão n.

A passagem para outra solução básica possível (vértice de poliedro de dimensão n) se faz através da troca de uma variável não-básica por uma básica com o aumento do valor da primeira (deslocamento sobre o eixo coordenado correspondente). Esse aumento é limitado pelas restrições do problema em relação ao valor desta variável. O limite é caracterizado por um vértice da região das soluções possíveis. Como a função objetivo Z é calculada em termos destas variáveis, procura-se variar aquela componente x_i que leve o valor da função objetivo o mais próximo do ótimo possível. Sempre que o valor da função objetivo for negativo ou zero para uma variável não-básica, significa que é possível realizar mais uma operação visando a uma otimização.

O número de iterações na solução de um problema de programação linear é função do problema que está sendo resolvido (natureza dos coeficientes, número de variáveis e restrições). Entretanto, é válido dizer que o número de iterações K está situado entre $1,5\ m \leq K \leq 2\ m$ onde m é o número de restrições[2,26]. Em cada iteração, são calculados $(m+1).(n+1)$ números com o que é válido afirmar que o presente método tem a desvantagem de exigir um grande número de operações para a sua aplicação.

Por outro lado, por ser um processo iterativo, os erros de arredondamento se acumulam através das iterações, podendo, inclusive, produzir resultados incorretos.

2.4 - Método "Simplex" com Multiplicadores

Tendo em vista os problemas apresentados acima, Dantzig[4] modificou o método "simplex" original de tal modo que os dados agora calculados e armazenados são chamados multiplicadores, e a matriz inversa da base obtida no sistema de equações originais.

Com isto, foram reduzidos os cálculos e diminuído o erro devido às iterações.

2.5 - Degeneração

Prova-se que o algoritmo "simplex" converge em um número finito de iterações, desde que os vetores escolhidos como básicos em cada iteração sejam linearmente independentes. Se os vetores forem linearmente dependentes, ocorre a degeneração, e o valor de Z pode não ser otimizado através de uma iteração. O problema de degeneração pode ser caracterizado pela situação em que um vértice do poliedro n-dimensional é determinado por mais de n vetores.

Para solucionar o problema de degeneração, Dantzig[4] desenvolveu o método "simplex" com perturbação, no qual um dos vetores linearmente dependentes é alterado ligeiramente de tal modo que modifica a determinação do vértice. O problema é resolvido com a perturbação, e a solução do problema é obtida fazendo esta perturbação desaparecer.

Se ocorre a degeneração, é possível, também, que uma base volte a aparecer no quadro "simplex"; é o chamado "cycling" no algoritmo "simplex" e resulta que não é possível chegar a uma solução ótima do problema.

Apesar de serem duas imperfeições do método, não constituem, na prática, grandes problemas, pois a degeneração pode ser evitada através de artifícios, e o "cycling" só ocorreu no método criado por Beale[4].

2.6 - Problema Dual

Seja o problema de programação linear 2.1 - 2.3 que será chamado de primal. O seu dual será definido como segue.

Problema Primal		Problema Dual
$\min Z = \sum_{j=1}^{n} c_j x_j$		$\max W = \sum_{i=1}^{m} (-b_i) y_i$
$\sum_{j=1}^{n} a_{ij} x_j + b_i \geq 0$	$i \in M_1$	$y_i \geq 0$
$\sum_{j=1}^{n} a_{ij} x_j + b_i = 0$	$i \in \overline{M}_1$	$y_i \gtrless 0$
$x_j \geq 0$	$j \in N_1$	$\sum_{i=1}^{m} a_{ij} y_i - c_j \leq 0$
$x_j \gtrless 0$	$j \in \overline{N}_1$	$\sum_{i=1}^{m} a_{ij} y_j - c_j = 0$

onde

$$M = \{1, \ldots. m\} \qquad M_1 = \{1, \ldots. m_1\} \qquad \overline{M}_1 = \{m_1+1, \ldots m\}$$

$$N = \{1, \ldots. n\} \qquad N_1 = \{1, \ldots. n_1\} \qquad \overline{N}_1 = \{n_1+1, \ldots n\}.$$

Associado ao problema dual, o teorema que segue é importante.

Teorema 2.7

a) Se o problema dual ou o primal tiver uma solução ótima finita, o outro também terá solução ótima finita de mesmo valor, isto é,

$$\min Z = \max W.$$

b) Se qualquer um dos problemas tiver solução ilimitada, o outro não terá solução. Ainda é possível ocorrer a situação em que nenhum dos dois tenha solução possível[4].

Uma conseqüência do teorema 2.7 é que sempre é possível escolher, entre o problema dual ou o primal, aquele que exige menor esforço para solução.

2.7 - IBM 1130 - Program LPMOSS

O Linear Programming Mathematical Optimization Subroutine System (LPMOSS) é um programa fornecido pela IBM para resolver problemas de programação linear no computador IBM 1130.

O algoritmo matemático utilizado é o método "simplex" com multiplicadores. O problema pode ser fornecido para a máquina através de cartões ou fita perfurada. O programa é armazenado no disco, e uma solução inicial anterior pode ser utilizada, quando gravada em disco, e aproveitada como solução inicial.

A capacidade de processamento é de problemas com 700 linhas, incluindo a função objetivo na forma matricial. A precisão depende do tamanho do problema bem como da grandeza dos coeficientes das variáveis.

Na resolução dos problemas-exemplo, foi utilizado um computador IBM 1130, tendo como unidade de entrada uma leitora de cartões e como unidade de saída uma impressora.

PARTE II

CAPÍTULO III

INSTABILIDADE DO LAÇO ALGÉBRICO EM COMPUTADOR ANALÓGICO

A resolução de problemas de programação matemática envolve, muitas vezes, operações puramente algébricas, principalmente quando particularizadas para a programação linear. Estas operações que formam um laço algébrico gerem instabilidades no computador analógico.

A Figura 3.1 mostra parte de um programa analógico hipotético onde aparecem K somadores numa malha fechada. Considerando a função de transferência de cada somador[20] como

$$F_i(s) = -\frac{G_i}{1 + \tau s},$$

a função de transferência da malha toda será[29]

$$\frac{V_{saída}}{V_{entrada}} = \prod_{i=1}^{n} F_i(s)$$

onde $$\frac{V_{saída}}{V_{entrada}} = \frac{(-1)^n G}{(1+\tau s)^n} \qquad \text{onde} \quad G = \prod_{i=1}^{n} G_i$$

A diferença de fase total para uma freqüência ω é

$$\Phi = n\left[\pi - tg^{-1}(\omega\,\tau)\right]$$

e a amplitude da função ganho

$$\left|\frac{V_{saída}}{V_{entrada}}\right| = \frac{G}{\left[1+(\omega\tau)^2\right]^{n/2}} \qquad 3.1$$

Para qualquer valor de ω, se a mudança de fase Φ for um múltiplo de 2π e o ganho maior do que a unidade, a malha será instável e a solução obtida não é a solução do problema.

Para atingir a estabilidade, a expressão 3.1 deve ter um valor menor do que 1, quando Φ for um número par de múltiplos de π radianos. É fácil ver que, se n é par e ω é igual a zero, a mudança de fase Φ é igual a um múltiplo de π radianos e o ganho à freqüência zero deve ser menor do que a unidade.

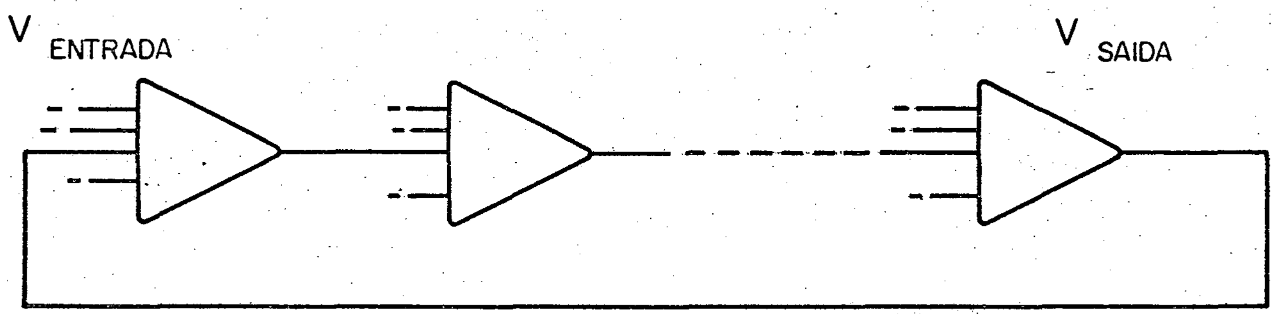


Figura 3.1

Quando n é ímpar, aparece realimentação negativa, e o ganho em freqüência zero poderá ser maior do que a unidade.

Se n é ímpar, uma mudança de fase de um múltiplo de 2π é dada pela expressão

$$\omega\tau = \text{tg}\,\frac{\pi}{n}\,.$$

O ganho a esta freqüência é dado por

$$\left|\frac{V_{saída}}{V_{entrada}}\right| = \frac{G}{\left[1+\text{tg}^2(\frac{\pi}{n})\right]^{n/2}} = G.\cos^n(\frac{\pi}{n})$$

Como este valor não pode ser maior do que a unidade devido a restrições de estabilidade, ou seja, de saturação, o máximo ganho G de laço é

$$G < \sec^n(\frac{\pi}{n}) .$$

Se n=3, o valor de G deverá ser sempre menor do que 8.

Devido a este problema, justifica-se pesquisar um método de transformar uma malha algébrica contendo somadores (inversores) apenas em malha diferencial contendo integradores (diferenciadores) para que seja possível resolver sistemas de equações ou desigualdades algébricas, ou melhor, resolver problemas que envolvam malhas puramente algébricas.

Um laço algébrico é caracterizado por uma equação algébrica

$$f(x) = 0$$

que, se for resolvida através do método da função implícita, resultará em um esquema semelhante ao da Figura 3.1.

Apresenta-se aqui, então, o método de contornar o problema de instabilidade, cujo resultado será posteriormente utilizado.

Como o presente trabalho se dirige para a programação linear, e ela envolve em geral um sistema de equações da forma

$$f_j(\underline{X}) = 0 \qquad (j=1, \ldots\ldots n)$$

onde $f_j(\underline{X})$ são funções lineares definidas no espaço n-dimensional, o método será desenvolvido para tal situação.

Tomando-se apenas uma equação

$$f(x) = 0 \qquad (x \text{ real}) \qquad 3.2$$

resolvida através da forma equivalente

$$x + K.f(x) = 0 \qquad (K >> 1) \qquad 3.3$$

de acordo com a Figura 3.2, sendo K o ganho do amplificador e pressupondo que o amplificador inverta o sinal, como um amplificador de alto ganho se comporta como um integrador com constante de tempo muito pequena, ou melhor, com um ganho de entrada muito grande, a equação 3.3 pode ser transformada em

$$\frac{dx}{dt} = -K\ f(x) \qquad 3.4$$

cujo circuito consta na Figura 3.3.

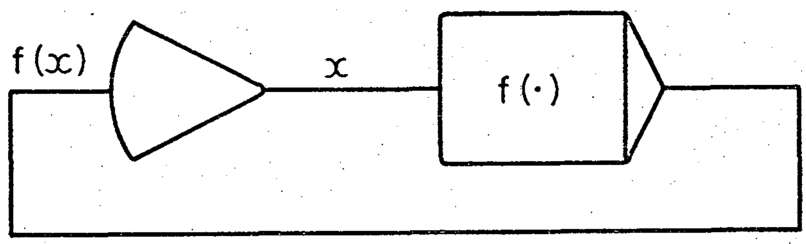


Figura 3.2

A solução do sistema será igual à do sistema 3.2 quando o ponto de equilíbrio $\frac{dx}{dt} = 0$ for estável.

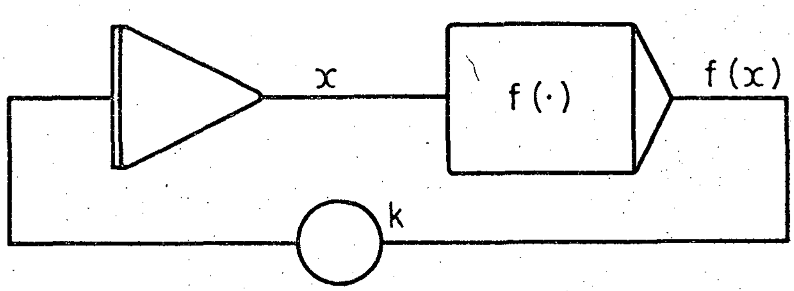


Figura 3.3

Como o método não pode ser limitado pela estabilidade do ponto de equilíbrio do sistema 3.4, procura-se, através da teoria da estabilidade de equações diferenciais, um sistema que tenha seu estado de equilíbrio estável e apresente a mesma solução do sistema 3.4.

Sendo dado o sistema de equações

$$f_j(x_1, \ldots x_n) = 0 \qquad (j=1, \ldots n) \qquad 3.5$$

onde x_i são as variáveis do problema, pressupõe-se que nas vizinhanças do ponto solução $\underline{X}^o$, ou seja,

$$\underline{X}^o = \begin{bmatrix} x_1^o \\ x_2^o \\ \vdots \\ x_n^o \end{bmatrix}$$

o determinante da matriz jacobiana seja diferente de zero, isto é,

$$|J(f\ \underline{X})| = \begin{vmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots\cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots\cdots & \frac{\partial f_n}{\partial x_n} \end{vmatrix}$$

para assegurar que o ponto de equilíbrio existe isoladamente.

O teorema de Liapunov sobre instabilidade assintótica permite formar um sistema de equações diferenciais cujo ponto de equilíbrio é solução de 3.5.

De acordo com um teorema de Liapunov, o ponto de equilíbrio $(\alpha_1, \alpha_2, \ldots \alpha_n)$ do conjunto de equações diferenciais

$$\frac{dy_i}{dt} = f_j(\alpha_1 + y_1, \ldots \alpha_n + y_n) \qquad (j=1, \ldots n)$$

é assintoticamente estável para valores muito pequenos de y_j(j=

=1, ... n), ou seja, na vizinhança do ponto $\underline{\alpha} = [\alpha_1,\ldots\alpha_n]$, se existe uma função V positiva definida $V = V(y_1,\ldots y_n)$, cuja derivada em relação a **t**, ao longo das trajetórias, é definida negativa, o que pode ser escrito na forma

$$\frac{dV}{dt} = \sum_{j=1}^{n} \frac{\partial V}{\partial y_i} \cdot \frac{dy_i}{dt} = \sum_{j=1}^{n} \frac{\partial V}{\partial y_i} \cdot f_j < 0 \qquad 3.6$$

Seja $V(\underline{Y})$ também uma função definida positivamente de f_1, $\ldots f_n$ ou

$$V = V(f_1,\ldots f_n)$$

onde

$$f_j = f_i(\alpha_1 + y_1,\ \alpha_2' + y_2,\ \ldots\ \alpha_n + y_n) \qquad (j=1,\ldots n)$$

a função $V(f_1,\ldots f_n) = 0$, no ponto $(x_1,\ldots x_n) = (\alpha_1,\ldots\alpha_n)$ onde $x_i = \alpha_i + y_i$ $(i=1,\ldots n)$, é uma solução de 3.5.

As características da função V são, então, as seguintes:

$$V > 0 \quad \text{quando} \quad f \neq 0 \qquad 3.7$$

$$V = 0 \quad \text{quando} \quad f = 0$$

e

$$\frac{dV}{dt} < 0 \quad \text{quando} \quad f \neq 0$$

$$\frac{dV}{dt} = 0 \quad \text{quando} \quad f = 0 \qquad 3.8$$

Tomando a expressão 3.6, vê-se que é equivalente a

$$\frac{dV}{dt} = \sum_{j=1}^{n} \frac{\partial V}{\partial x_j} \cdot \frac{dx_j}{dt} \qquad 3.9$$

Para formar um sistema de equações diferenciais cujo ponto de equilíbrio seja solução de 3.5 e seja assintoticamente estável, far-se-á

$$\frac{dx_j}{dt} = -K_j \frac{\partial V}{\partial x_j} \qquad K_j > 0 \qquad 3.10$$

o que garante que

$$\frac{dV}{dt} = -\sum_{j=1}^{n} K_j \left(\frac{\partial V}{\partial x_j}\right)^2 \leq 0 \qquad 3.11$$

Mas as condições 3.8 ainda não estão plenamente satisfeitas; pode existir um ponto tal que

$$f \neq 0 \quad \text{e} \quad \frac{dV}{dt} = 0 \qquad 3.12$$

Para $\frac{dV}{dt} = 0$, tem-se

$$\frac{\partial V}{\partial x_j} = \sum_{k=1}^{n} \frac{\partial f_k}{\partial x_j} \cdot \frac{\partial V}{\partial f_k} = 0,$$

ou seja,

$$\frac{\partial V}{\partial \underline{X}} = J^T \frac{\partial V}{\partial \underline{f}} = \underline{0}. \qquad 3.13$$

Para que 3.13 seja verdadeira pela teoria das matrizes se afirma que[11], ou

$$|J^T| = 0, \qquad \text{ou} \qquad \frac{\partial V}{\partial \underline{f}} = \underline{0}.$$

Supondo que

$$\frac{\partial V}{\partial \underline{f}} = \underline{0}$$

somente quando $f(\underline{X}) = 0$, a única situação em que 3.12 se verifica é quando

$$|J\, [\, \underline{f} \quad \underline{X}\,]| = 0, \qquad \text{ou seja,}$$

a matriz jacobiana é singular o que contraria a hipótese inicial. Então, 3.11 é sempre negativa e $\frac{dV}{dt}$ é definida negativa.

Estes pontos, ou são instáveis, ou são falsos pontos de equilíbrio, chamados mínimos locais[11]. Não são solução do sistema 3.5 o que pode ser facilmente detectado através da expressão

$$\sum_{j=1}^{n} |f_j| \neq 0.$$

Uma vez que

$$\frac{d\underline{X}}{dt} = -\underline{K}\,\frac{\partial V}{\partial \underline{X}}$$

e

$$\frac{\partial V}{\partial \underline{X}} = J^T\,\frac{\partial V}{\partial \underline{f}}$$

onde $\underline{K}$ é uma matriz ganho com $K_{ij} \neq 0$ para $i,j=1,\ldots n$, pode-se escrever

$$\frac{d\underline{X}}{dt} = -\underline{K}\;J^T\,\frac{\partial V}{\partial \underline{f}}$$

Obedecidas às restrições 3.7 e 3.8, obteve-se um conjunto de equações diferenciais

$$\frac{d\underline{X}}{dt} + \underline{K}\,J^T\,\frac{\partial V}{\partial \underline{f}} = \underline{0}$$

cuja solução estável coincide com a solução do sistema de equações

$$f_j(\underline{X}) \equiv 0 \qquad (j=1,\ldots n)$$

As formas mais comuns para a função de Liapunov são:

$$V = \frac{1}{2}\sum_{k=1}^{n} f_k^{\,2}\,(\underline{X}) \tag{3.14}$$

ou

$$V = \sum_{k=1}^{n} |f_k(\underline{X})|$$

Utilizando a forma 3.14 para um sistema de equações lineares do tipo

$$f_i(\underline{X}) = \sum_{j=1}^{n} a_{ij}x_j + b_i = 0 \qquad (i=1,\ldots n),$$

o sistema de equações diferenciais equivalente será

$$\frac{dx_j}{dt} + K_j\sum_{k=1}^{n} a_{ik}\,(\sum_{p=1}^{n} a_{kp}x_p + b_k) = 0$$

Pode-se sugerir um método de solução do problema de programação linear em computador analógico levando em consideração os teoremas 2.4, 2.5 e 2.6. Este método consiste em pesquisar entre as soluções das combinações de n equações entre as m restrições do problema linear aquela que otimiza a função objetivo.

O método não será explorado, porque é bem claro que envolve muitas tentativas.

Por outro lado, é com base no teorema de Liapunov que se poderá afirmar a convergência do método do gradiente no próximo capítulo.

CAPÍTULO IV

SOLUÇÃO DE PROBLEMAS DE PROGRAMAÇÃO CONVEXA EM COMPUTADOR ANALÓGICO

O método de solução aqui apresentado consiste em reduzir um problema de mínimo condicionado ao de encontrar o mínimo absoluto de uma determinada função auxiliar através do chamado método da função penalidade ("penalty function").

Em problemas de programação convexa, o ponto de mínimo da função auxiliar não coincide com o mínimo do problema original, mas, nos problemas de programação linear, é possível construir uma função penalidade que tenha os mesmos pontos de mínimo.

Aqui será montado um sistema de equações diferenciais cuja solução, independentemente das condições iniciais, convergirá para os pontos de mínimo não-condicionado.

Inicialmente, serão apresentados os fundamentos do método para os problemas de programação convexa que, a seguir, serão particularizados para programação linear.

4.1 - Formulação do Problema[22]

Seja a função objetivo, convexa, duas vezes continuamente diferenciável

$$F(\underline{X}) = F(x_1, \dots x_n) \qquad 4.1$$

que se quer minimizar sujeita às restrições

$$\phi_i(\underline{X}) \leq 0 \qquad (i=1, \ldots m) \qquad 4.2$$

em que $\underline{X}$ pertence ao espaço euclidiano de dimensão n e $\phi_i(\underline{X})$ é uma função convexa definida em todo o espaço e duas vezes continuamente diferenciável. As restrições do tipo

$$x_j \geq 0 \qquad (j=1, \ldots n),$$

quando consideradas na programação convexa, podem estar incluídas em 4.2 sob a forma de

$$-x_j \leq 0.$$

Juntamente com 4.2, pode ser dado o conjunto de equações lineares simultâneas

$$f_s(\underline{X}) = \sum_{k=1}^{q} a_{sk}x_k + b_s = 0 \qquad (s=1,\ldots q),(q < n) \qquad 4.3$$

onde a_{sk} e b_s são números reais.

Seja G o conjunto de pontos, que é convexo, o qual satisfaz simultaneamente aos sistemas 4.2 e 4.3, supondo que o conjunto G não é vazio e que a solução do problema é obtida em um único ponto da fronteira de G, pesquisar-se-á a solução exata do problema auxiliar e a solução aproximada do problema dado.

4.2 - Construção da Função Penalidade[22]

Seja D a região definida pelas desigualdades 4.2. Construindo uma função convexa $L(\underline{X}) = L(x_1,\ldots x_n)$ tal que $L(\underline{X}) = 0$ para $\underline{X} \in D$ e $L(\underline{X}) > 0$ para $\underline{X} \notin D$, pode-se formar

$$L(\underline{X}) = \sum_{i=1}^{m} \mu_i \, H_i\left[\phi_i(\underline{X})\right] \qquad \mu_i > 0$$

onde $H_i\left[\phi_i(\underline{X})\right]$ é convexa no espaço n-dimensional e continuamente diferenciável definida como segue:

$$H_i\left[\phi_i(\underline{X})\right] \begin{cases} \theta i\left[\phi_i(\underline{X})\right] > 0 & \text{para} \quad \phi_i(\underline{X}) > 0 \\ 0 \quad \text{para} \quad \phi_i(\underline{X}) \leq 0 . \end{cases} \qquad 4.4$$

A função $H_i\left[\phi_i(\underline{X})\right]$ pode ser da forma

$$H_i\left[\phi_i(\underline{X})\right] = \frac{1}{2}\,\phi_i{}^2(\underline{X}) \quad \delta\phi_i(\underline{X}) \qquad (i=1,\ldots m) \qquad 4.5$$

onde

$$\theta_i\left[\phi_i(\underline{X})\right] = \frac{1}{2}\,\phi_i{}^2(\underline{X})$$

$$\delta\phi i(\underline{X}) \begin{cases} 1 & \text{se } \phi_i(\underline{X}) > 0 \\ 0 & \text{se } \phi_i(\underline{X}) \leq 0. \end{cases}$$

A função $L(\underline{X})$ tem valor zero somente em D onde todas as restrições estão satisfeitas. Fora de D, existirá pelo menos uma desigualdade que não estará satisfeita e, então, $L(\underline{X}) > 0$. A função $L(\underline{X})$ é convexa porque é soma de funções convexas com coeficientes positivos.

Para a pesquisa dos pontos pertencentes à região E definida pelas equações 4.3, pode-se tomar a função M na forma

$$M(\underline{X}) = \frac{1}{2}\sum_{i=1}^{q}\gamma_i\; f_i^2(\underline{X}) \qquad \gamma_i > 0 \qquad 4.6$$

A função $M(\underline{X})$ é convexa, uma vez que $f_i(\underline{X})$ $(i=1,\ldots q)$ são funções lineares; aqui, também, $M(\underline{X})$ só será nula quando pertencer à região E.

Os pontos que pertencem à região G, isto é, $D \cap E$, são encontrados através da função

$$U(\underline{X}) = L(\underline{X}) + M(\underline{X}) \qquad 4.7$$

a qual tem a mesma propriedade de suas componentes e que assume valores em R.

Seja, agora, a função penalidade que é convexa por construção

$$W(\underline{X},\alpha) = F(\underline{X}) + \alpha\, U(\underline{X}). \qquad 4.8$$

Define-se $\pi(\alpha)$ como o conjunto de pontos de mínimo tais que

$$\pi(\alpha) = \{\underline{\overline{X}} | W(\underline{\overline{X}},\alpha) \leq W(\underline{X},\alpha),\ \forall\ \underline{X} \in G\}$$

e

$$\pi(\alpha') = \{\underline{\overline{X}},\alpha | W(\underline{\overline{X}},\alpha) \leq W(\underline{\overline{X}},\alpha'),\ \forall\ \underline{X} \in G \ \text{ e } \ \alpha \geq \alpha'\}.$$

Para a função do tipo 4.8, prova-se que existe um mínimo.

Teorema 4.1

Sejam as funções $F(\underline{X})$, $U(\underline{X})$ e $W(\underline{X},\alpha) = F(\underline{X}) + \alpha\, U(\underline{X})$ convexas e definidas em E_n ($\underline{X} \in E_n$), onde $U(\underline{X}) = 0$, num conjunto convexo G e seja o valor mínimo de $F(\underline{X})$ ($\underline{X} \in G$) atingido na fronteira de G em um único ponto $\underline{X} = \underline{X}^o$. Então:

1 - existe um número α' tal que, para qualquer $\alpha \geq \alpha'$, a função $W(\underline{X},\alpha)$ atinge um mínimo;

2 - dados γ e ξ arbitrários, existe um α'', $\alpha'' = \alpha''(\gamma,\xi)$, $\alpha'' \geq \alpha'$, tal que, para qualquer $\alpha > \alpha''$, em cada ponto $\underline{X} = \underline{\overline{X}}$, ($\underline{\overline{X}} \in \pi(X)$) é verdade

$$\max_i \{|\overline{x}_i - x_i^o|\} \leq \gamma$$

e

$$|W(\underline{\overline{X}},\alpha) - F(\underline{X}^o)| \leq \xi$$

onde $\overline{x}_i$ e x_i^o são as componentes $\underline{\overline{X}}$ e $\underline{X}^o$ respectivamente. A prova do teorema se encontra no Anexo I.

De acordo com ele, para valores muito grandes do parâmetro α, a função $W(\underline{X},\alpha)$ tem um mínimo absoluto perto do ponto $\underline{X}^o$ que é o ponto de solução ótima do problema 4.1-4.3. O conjunto de pontos de mínimo de $W(\underline{X},\alpha)$ forma um conjunto $\pi(\alpha)$, e suas componentes não diferem das componentes x_i por mais do que um dado valor γ, e o valor da função nos pontos de mínimo não difere do valor ótimo por mais do que um dado número ξ. Quando α tende para infinito, o valor de $W(\underline{X},\alpha)$ tende para o valor mínimo (ótimo) de $F(\underline{X})$.

Não existe método geral de determinação de α" dados γ e ξ. O referido teorema tem caráter existencial; entretanto, fornece base teórica para selecionar um α muito grande o que garante um erro pequeno.

A pesquisa do mínimo absoluto será feita através do método do gradiente desta função com a investigação da convergência.

4.3 - Método do Sistema Gradiente para Minimização

Introduzindo o sistema gradiente de equações diferenciais

$$\tau_i \frac{dx_i}{dt} = - \frac{\partial S(\underline{X})}{\partial x_i} \qquad \tau_i > 0 \ (i=1,\dots n) \qquad 4.9$$

onde a função $S(\underline{X})$ é definida em E_n e continuamente diferenciável, de acordo com estas equações, as componentes de velocidade de fase $\frac{dx_i}{dt}$ são proporcionais às componentes do vetor gradiente tomadas com sinal inverso.

Com relação ao sistema 4.9, vale afirmar que, se a função $S(\underline{X})$ é convexa e atinge um mínimo no conjunto limitado σ, então a solução do sistema converge globalmente para este conjunto. Isto significa dizer que, se $S(\underline{X})$ é estritamente convexa, o sistema tem pontos de equilíbrio assintoticamente estáveis, ou a função atinge um mínimo num único ponto definido pelo sistema de equações

$$\frac{\partial S(\underline{X})}{\partial x_i} = 0 \qquad (i=1, \ldots n).$$

Dados γ e ξ, escolhido um $\alpha > \alpha''$ o que assegura a existência do mínimo da função $W(\underline{X},\alpha)$ e sendo verdadeiras as afirmações do teorema 4.1, pode-se considerar o seguinte sistema de equações diferenciais para o método do gradiente

$$\tau_i \frac{dx_i}{dt} = - \frac{W(\underline{X},\alpha)}{x_i} \qquad \tau_i > 0 \qquad (i=1,\ldots n) \qquad 4.10$$

o que permite afirmar:

Teorema 4.2[22]

Para γ e ξ positivos, muito pequenos, existe um α'' tal que, para $\alpha \geq \alpha''$, a solução do sistema converge para um subconjunto de soluções ótimas (γ,ξ) do problema.

Substituindo $W(\underline{X},\alpha)$ em 4.10 por sua forma derivada e em função de $F(\underline{X})$ e $U(\underline{X})$, o sistema pode ser escrito na forma

$$\tau_i \frac{dx_i}{dt} = -\frac{\partial F(\underline{X})}{\partial x_i} - \alpha \left[\sum_{k=1}^{m} \phi_k(\underline{X}) \frac{\partial \phi_k(\underline{X})}{\partial x_i} \delta\phi_k(\underline{X}) + \sum_{s=1}^{q} f_s(\underline{X}) \frac{\partial f_s(\underline{X})}{\partial x_i} \right]$$
$$(i=1,\ldots n) \qquad 4.11$$

O sistema 4.11 é aquele que será modelado no computador. Para as condições iniciais e um valor α muito grande, a solução $\underline{X}(\underline{X}, t)$ tenderá para um ponto de equilíbrio; se, com maior aumento de α, não existir grande variação em $W(\underline{X},\alpha)$ ou nas componentes do ponto de repouso, pode-se afirmar que a solução ótima foi atingida.

4.4 - Programação Linear

O problema de programação linear, como já foi dito, consiste em minimizar a forma linear

$$Z(\underline{X}) = \sum_{j=1}^{n} c_j x_j \qquad 4.12$$

encontrando um ponto ótimo

$$\underline{X}^o = \left[x_1^o, x_2^o, \ldots . x_n^o \right]$$

pertencente a uma região G definida por

$$
\begin{aligned}
f_i(\underline{X}) &= \sum_{j=1}^{n} a_{ij} x_j + b_i \leq 0 && (i=1, \ldots m) \\
\phi_s(\underline{X}) &= -x_s \leq 0 && (s=1, \ldots n) \qquad 4.13 \\
f_t(\underline{X}) &= \sum_{j=1}^{n} a_{tj} x_j + b_t = 0 && (t=1, \ldots p)\,(p<n)
\end{aligned}
$$

A forma mais usual da função penalidade, para este caso, é

$$W(\underline{X},\alpha) = Z(\underline{X}) + \alpha \left[\sum_{i=1}^{m} f_i^2(\underline{X})\ \delta f_i(\underline{X}) + \sum_{s=1}^{n} x_s^2\ \delta(-x_1) + \sum_{t=1}^{p} f_t^2(\underline{X}) \right] \qquad 4.14$$

onde

$$\delta f_i(\underline{X}) \begin{cases} 0 & \text{se} \quad f_i(\underline{X}) \leq 0 \\ 1 & \text{se} \quad f_i(\underline{X}) > 0 \end{cases}$$

e $\alpha > 0$ é um parâmetro.

As restrições da forma de equações podem ser transformadas em desigualdades através das equivalências caracterizadas no Capítulo II.

O sistema gradiente para a função penalidade 4.12 é da forma

$$-\tau_i \frac{dx_j}{dt} = +c_j + 2\alpha \left[\sum_{i=1}^{m} a_{ij} f_i(\underline{X}) \, \delta f_i(\underline{X}) + x_j \, \delta(-x_j) + \sum_{t=1}^{p} a_{tj} \, f_t(\underline{X}) \right]$$

$$(j=1,\ldots n) \qquad 4.15$$

Novamente se diz que existe um valor α_o tal que, para $\alpha > \alpha_o$, a função $W(\underline{X},\alpha)$ tem um mínimo absoluto e, quando α tende para infinito, o conjunto de pontos que verificam o mínimo, de acordo com os teoremas 4.1 e 4.2, tende para a solução $\underline{X}^o$ do problema de programação linear.

A precisão do problema depende de α. Em problemas reais, em lugar de apenas um coeficiente α, introduzem-se outros dois coeficientes β e $\overline{\alpha}$ tais que $\frac{\overline{\alpha}}{\beta} = \alpha$. E, em lugar de se trabalhar com a função $W(\underline{X},\alpha)$, considera-se a nova função

$$W(\underline{X},\beta,\overline{\alpha}) = \beta \, Z(\underline{X}) + \overline{\alpha} \, U(\underline{X})$$

para um $\overline{\alpha}$ muito grande e β pequeno ($\overline{\alpha} >> \beta$) obtém-se uma solução mais exata.

É possível constituir uma outra função penalidade cujo mínimo coincide com a solução do problema de programação linear .

Seja a função $W(\underline{X},\alpha)$ na forma

$$W(\underline{X},\alpha) = \sum_{j=1}^{n} c_j x_i + \alpha \left[\sum_{i=1}^{m} f_i(\underline{X}) \; \delta f(\underline{X}) + \sum_{s=1}^{n} x_s \, \delta(-x_s) + \sum_{t=1}^{p} f_t(\underline{X}) \right]$$

Ainda assim, $W(\underline{X},\alpha)$ é convexa e contínua, mas não indiferenciável em todo o espaço, pois não é definida para $f_i(\underline{X})=0$ ou $x_s=0$. Porém, permite provar o seguinte teorema

Teorema 4.3

Se o problema de programação linear atinge a solução ótima em um único ponto $\underline{X}=\underline{X}^o$, pode-se encontrar um $\bar{\alpha} > 0$, tal que, para todo $\alpha > \bar{\alpha}$, a função $W(\underline{X},\alpha)$ definida em 4.16 tem o mínimo em $\underline{X}=\underline{X}^o$. A prova deste teorema pode ser vista na forma do presente enunciado em "The gradient method of solving linear and quadratic programming problems on electronic analog computers", de Ribashov[23].

Para encontrar o mínimo desta função, utiliza-se o sistema gradiente de equações diferenciais da forma

$$\frac{dx_j}{dt} = -\frac{1}{\tau_j}\frac{\partial W(\underline{X},\bar{\alpha})}{\partial x_j} \qquad (j=1, \ldots n) \tag{4.17}$$

onde $\tau_j > 0$ $(j=1,\ldots n)$ e as derivadas parciais com $\delta f_i(\underline{X})$ = constante $(i=1, \ldots m)$.

O sistema acima representa descontinuidades de primeira espécie quando $\phi_s = 0$ e $f_i = 0$ e em suas interseções.

Examinando a derivada total

$$\frac{dW}{dt} = \sum_{j=1}^{n} \frac{\partial W(\underline{X},\alpha)}{\partial x_j} \cdot \frac{dx_j}{dt} \tag{4.18}$$

vê-se que é negativa fora da região M onde ocorrem as descontinuidades. Se a condição inicial é tomada fora da região M pelo gradiente, a função diminuirá até atingir M; então, dois casos podem acontecer: ou a trajetória atravessa a região M e segue pesquisando o mínimo de $W(\underline{X},\alpha)$, ou permanece na região M no "slipping mode", uma vez que $\frac{dW}{dt} < 0$ em todo o lugar fora de M. Assim, haverá uma oscilação na vizinhança de $\underline{X}^o$. O método de Pyne[19] é uma particularização deste sistema.

O sistema gradiente dado pela forma 4.17 não será estudado em detalhe neste trabalho, tendo em vista a desvantagem da oscilação que pode apresentar.

4.5 - Circuitos para o Computador Analógico

Para a realização das funções de minimização no computador analógico, além dos elementos lineares (integradores, somadores, etc.), é necessária a representação de uma função não-linear $\delta f_i(\underline{X})$ da forma

$$\delta f_i(\underline{X}) \begin{cases} 1 & \text{se} \quad f_i > 0 \\ 0 & \text{se} \quad f_i \leq 0 \end{cases}$$

e outra da mesma forma, mas que representa a restrição de não-negatividade das variáveis x, ou seja,

$$\delta(-x_i) \begin{cases} 0 & \text{se} \quad x_i \geq 0 \\ 1 & \text{se} \quad x_i < 0 \end{cases}$$

Tal função pode ser realizada através dos seguintes modos :

- com relés eletromecânicos;
- com comparadores eletrônicos a diodo;
- com sub-rotinas especiais.

Os relés eletromecânicos têm a desvantagem de serem lentos e apresentarem perdas em sua atuação.

O comparador eletrônico, que é fornecido pelo próprio fabricante do computador analógico, funciona da seguinte maneira de acordo com a Figura 4.1.

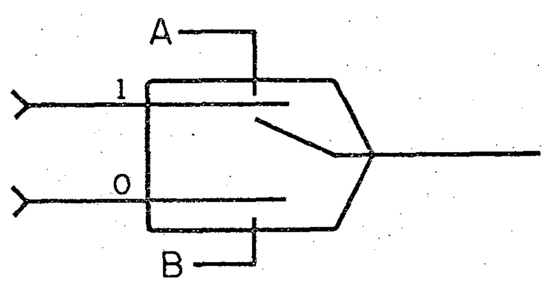


Figura 4.1.

Se $A + B > 0$, a chave vai para a posição 1 e,se $A + B < 0$, a chave toma a posição 0.

Então, o termo $f_i . \delta f_i$, na expressão 4.15, é representado através de um bloco analógico como se apresenta na Figura 4.2.

Tomando $e < 0$

. se $f_i > |e|$, $f_i + e > 0$ e o comparador atua na posição 1,

. se $f_i = 0$, $f_i + e < 0$ e o comparador atua na posição 0,

. se $f_i < 0$, $f_i + e < 0$ e o comparador atua na posição 0,

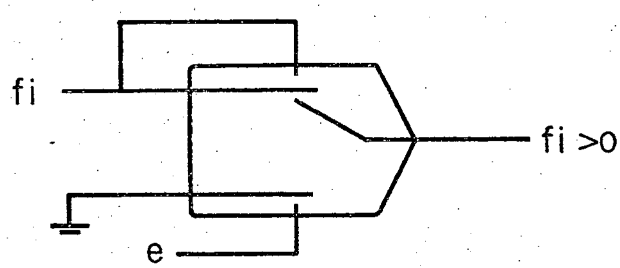


Figura 4.2

O sinal e negativo representa um erro na fixação da restrição f_i, ou seja, no caso, está-se fazendo $f_i \leq e$.

O mesmo é feito quanto ao fator $x_i . \delta(-x_i)$ de acordo com a Figura 4.3. Neste caso, o sinal e é positivo, pois representa $x_i \geq e$.

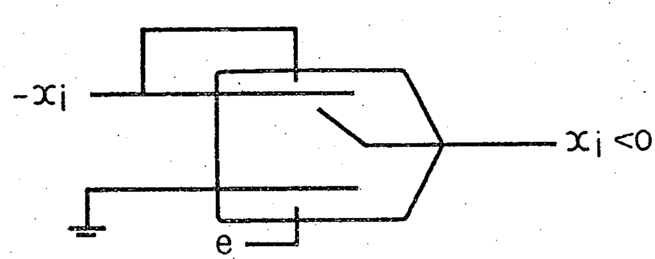


Figura 4.3

Outra forma de realizar um comparador é utilizar um amplificador operacional realimentado através de diodo conforme Figura 4.4.

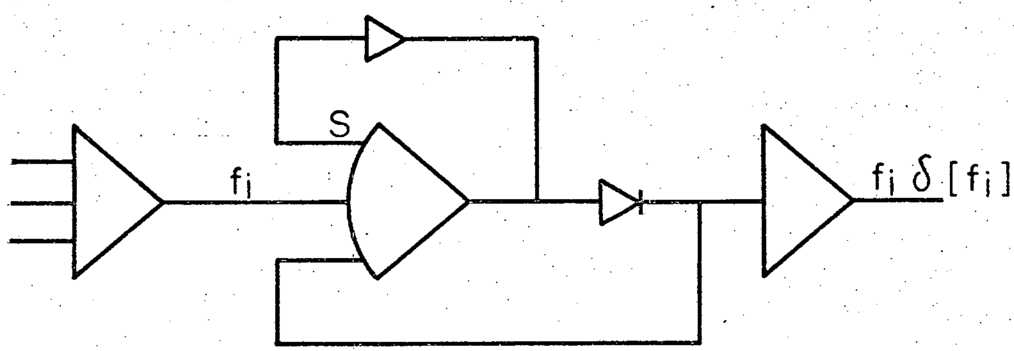


Figura 4.4

Os inversores na saída e entrada do conjunto comparador são para evitar o carregamento do circuito através de impedâncias de entrada muito baixas.

Outros esquemas foram abandonados por apresentarem as desvantagens de serem instáveis[5] ou por serem muito complicados[28,11].

A forma genérica do circuito com comparadores para o computador analógico do sistema 4.15 é apresentado na Figura 4.5.

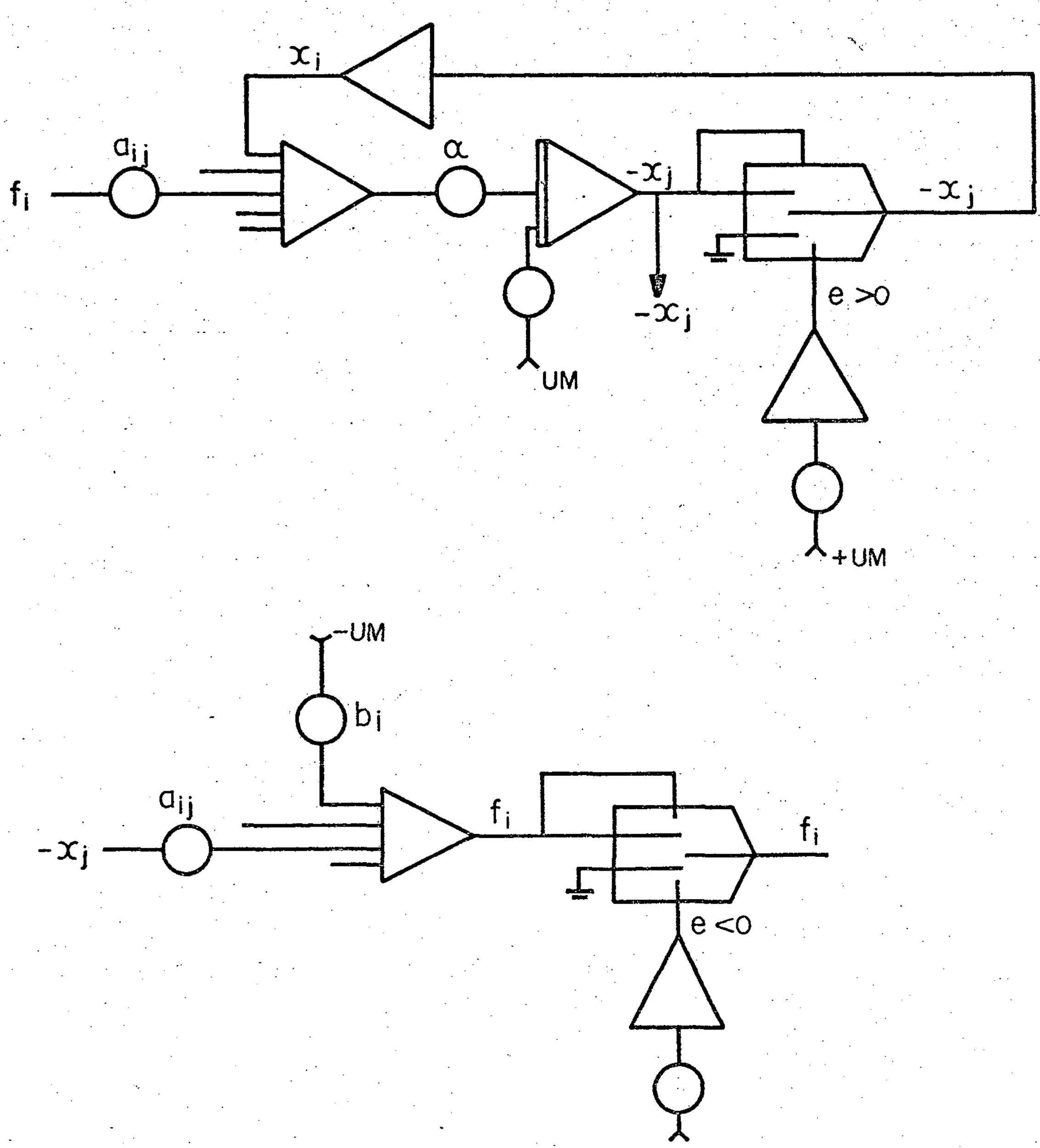


Figura 4.5

Se forem utilizados os circuitos com diodos externos, a diagramação será a da Figura 4.6.

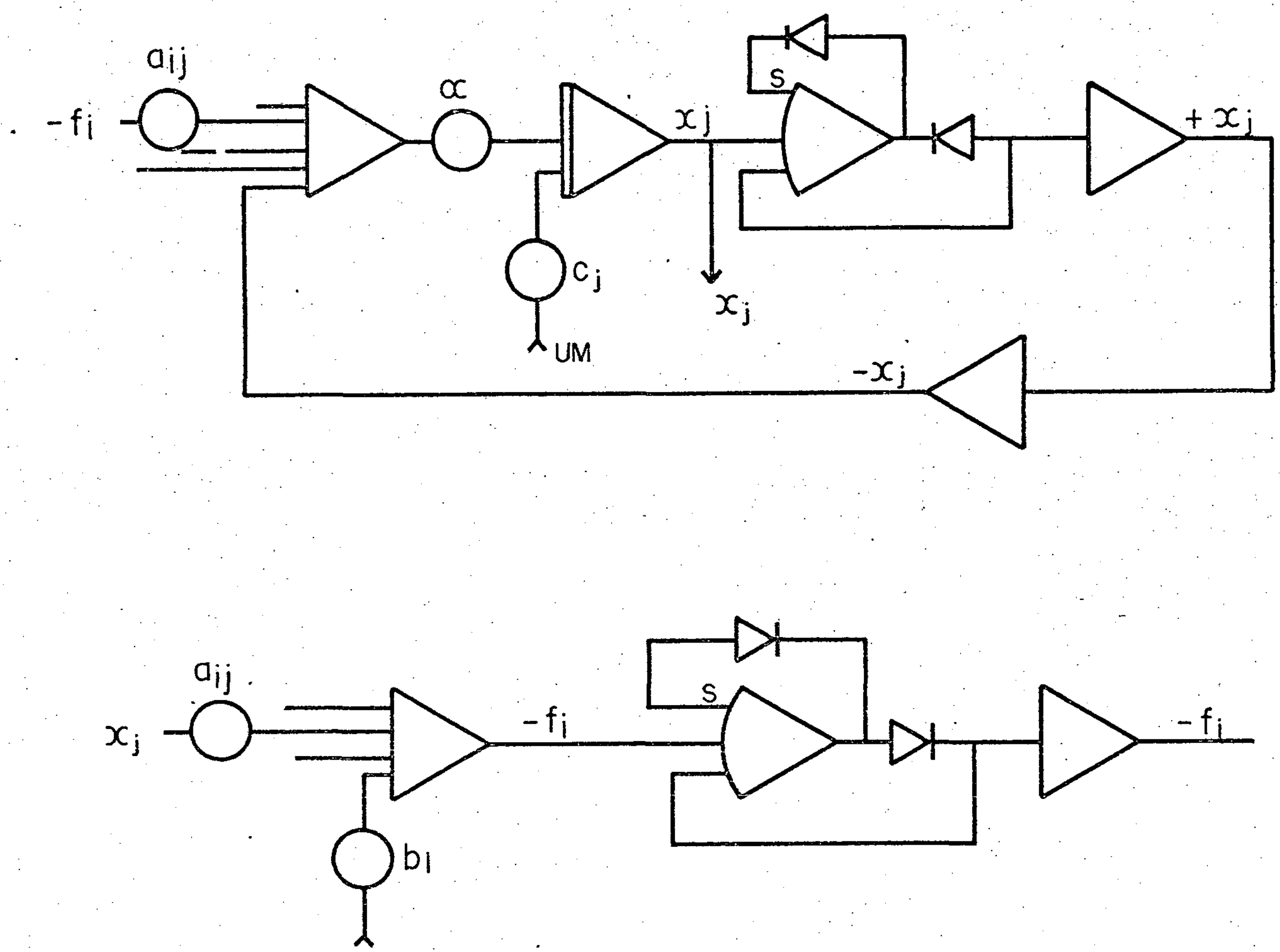


Figura 4.6.

CAPÍTULO V

EXEMPLOS

Os exemplos que aqui serão resolvidos, visam a comprovar o método. Foram escolhidos na bibliografia consultada e, eventualmente, será feita alguma observação sobre particularidades dos mesmos.

A metodologia de solução será a seguinte:

a) apresentação do problema na sua forma usual;

b) obtenção da solução analógica:

   b.1) colocação do problema na forma adequada;

   b.2) escalonamento para o computador analógico;

   b.3) formulação das equações do sistema gradiente;

   b.4) apresentação do diagrama analógico;

   b.5) apresentação dos resultados;

c) apresentação dos resultados digitais, detalhados no Anexo II;

d) comentários.

EXEMPLO 1

a) Dado o problema

minimizar $Z = 0,4x_1 + 1,6x_2$

sujeito às restrições

$$1,8x_1 + 0,175x_2 \leq 1$$
$$-1,5x_1 + x_2 \leq 1$$
$$x_1 \geq 0$$
$$x_2 \geq 0$$

b.1) Colocando o problema na forma anteriormente apresentada

$$\text{MIN } Z = 0,4x_1 + 1,6x_2$$
$$f_1 = 1,8x_1 + 0,175x_2 - 1 \leq 0$$
$$f_2 = -1,5x_1 + x_2 - 1 \leq 0$$
$$\phi_1 = -x_1 \leq 0$$
$$\phi_2 = -x_2 \leq 0.$$

b.2) Escalonando as variáveis de forma a evitar a saturação, fazendo

$$x_{i_{max}} = 10 \qquad \text{ou} \qquad y_i = \frac{x_i}{10},$$

tem-se

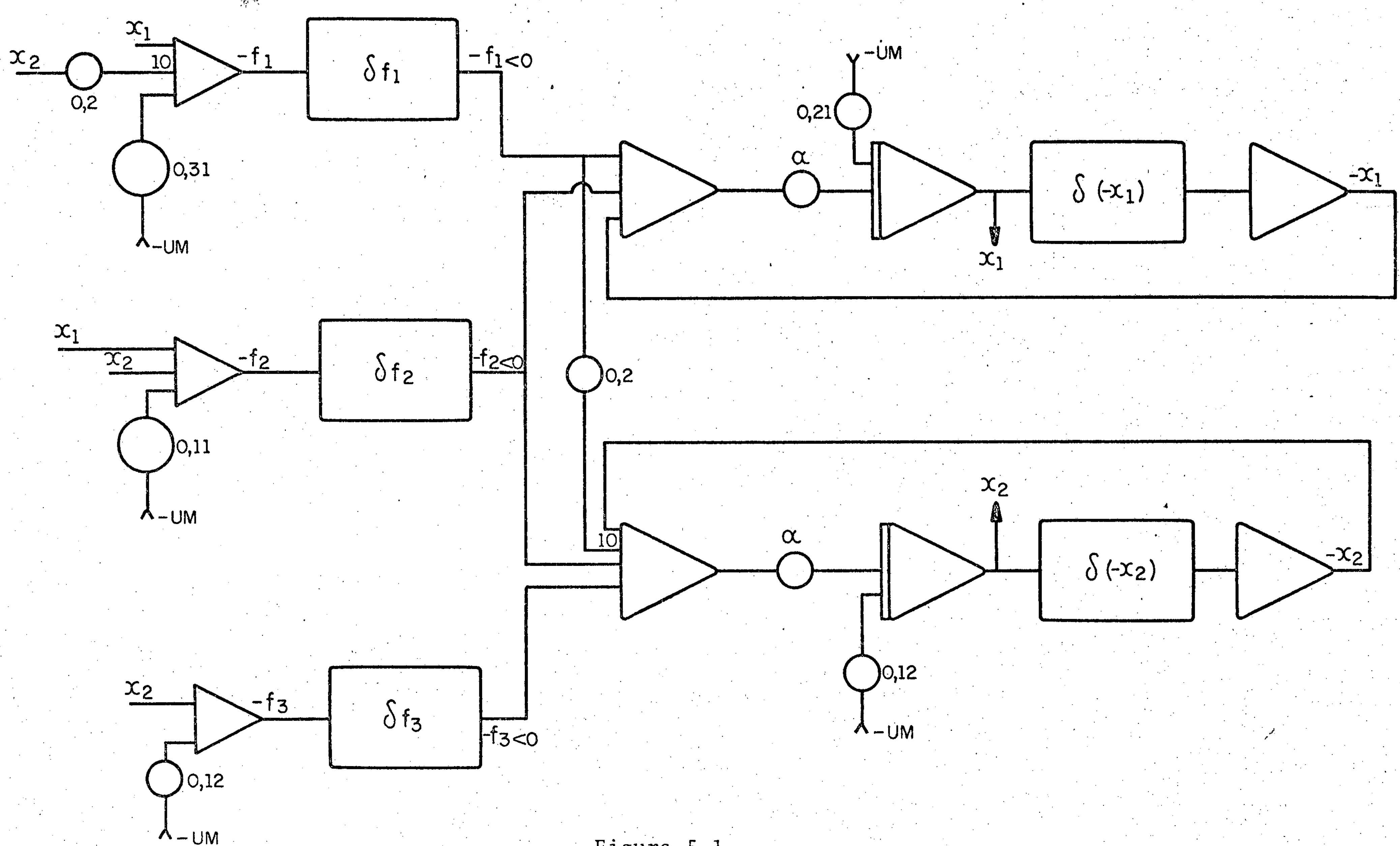
$$\text{MIN } Z = 0,4y_1 + 1,6y_2$$
$$f_1 = 1,8y_1 + 0,175y_2 - 0,1 \leq 0$$
$$f_2 = -1,5y_1 + y_2 - 0,1 \leq 0$$
$$\phi_1 = -y_1 \leq 0$$
$$\phi_2 = -y_2 \leq 0$$

b.3) O sistema gradiente terá a forma

$$-\frac{dy_1}{dt} = 0,4 + \alpha\left[1,8f_1\ \delta f_1 - 1,5\ f_2\ \delta f_2 + y_1\delta(-y_1)\right]$$

$$-\frac{dy_2}{dt} = 1,6 + \alpha\left[0,175f_1\ \delta f_1 + f_2\ \delta f_2 + y_2\ \delta(-y_2)\right]$$

b.4) A Figura 5.1 mostra o circuito analógico utilizado.



x2
x1
10
0,2
-f1
δf1
-f1<0
0,31
-UM
-UM
0,21
α
δ(-x1)
-x1
x1
x1
x2
-f2
δf2
-f2<0
0,11
-UM
0,2
x2
10
α
δ(-x2)
-x2
0,12
-UM
x2
-f3
δf3
-f3<0
0,12
-UM

Figura 5.1.

b.5) O quadro apresenta os resultados obtidos para as diversas condições iniciais introduzidas e os diversos valores de α.

α	CONDIÇÕES INICIAIS		RESULTADOS		TEMPO(S)	
	y_1	y_2	y_1	y_2	y_1	y_2
1	0	0	0,2785	0,2369	2	2
1	-0,5	-0,5	0,2785	0,2369	3	5
1	-0,99	-0,99	0,2785	0,2369	4	8
10	0	0	0,0402	0;0159	2	2
10	-0,5	-0,5	0,0402	0,0159	3	5
10	-0,99	-0,99	0,0402	0,0159	4	8
20	0	0	0,0203	0,0080	2	2
20	-0,5	-0,5	0,0203	0,0080	3	5
20	-0,99	-0,99	0,0203	0,0080	4	8
100	0	0	0,0042	0,0015	2	2
100	-0,5	-0,5	0,0042	0,0015	3	5
100	-0,99	-0,99	0,0042	0,0015	4	8
200	0	0	0,0021	0,0007	2	2
200	-0,5	-0,5	0,0021	0,0007	3	5
200	-0,99	-0,99	0,0021	0,0007	4	8
1 000	0	0	0,0006	0,0002	2	2
1 000	-0,5	-0,5	0,0006	0,0002	3	5
1 000	-0,99	-0,99	0,0006	0,0002	4	8

A melhor solução obtida no computador analógico foi para um valor de α = 1 000 e

$y_1 = 0,0006$

$y_2 = 0,0002$

donde

$x_1 = 0,06$

$x_2 = 0,02$

MIN Z = 0,056

quando os valores ótimos são

$x_1 = 0$

$x_2 = 0$

MIN Z = 0

c) O problema resolvido da mesma forma no computador IBM 1130, utilizando o programa LPMOSS[13], apresentou o seguinte resultado:

$x_1 = 0,0$

$x_2 = 0,0$

MIN Z = 0,0

sendo que o tempo de solução foi de 10 minutos.

d) Comentários.

O tempo neste problema foi marcado por um contador de segundos montado na parte lógica do computador Telefunken RA 770. Portanto, é suficientemente preciso para dar uma idéia da ordem de grandeza do tempo de convergência do método.

Pôde-se constatar que:

- o tempo de convergência para a situação estável é quase só função da condição inicial;
- a precisão, ou melhor, a convergência para o ponto mínimo é função do valor de α.

EXEMPLO 2

a) Dado o problema

maximizar $Z = 21x_1 + 12x_2$

sujeito às restrições

$$x_1 + 2x_2 \leq 31$$
$$x_1 + x_2 \leq 11$$
$$x_2 \leq 12$$
$$x_1 \geq 0$$
$$x_2 \geq 0.$$

b.1) Colocando o problema na forma anteriormente apresentada

$$\text{MIN } Z = -21x_1 - 12x_2$$
$$f_1 = x_1 + 2x_2 - 31 \leq 0$$
$$f_2 = x_1 + x_2 - 11 \leq 0$$
$$f_3 = x_2 - 12 \leq 0$$
$$\phi_1 = -x_1 \leq 0$$
$$\phi_2 = -x_2 \leq 0.$$

b.2) Escalonando as variáveis de forma a evitar a saturação, fazendo

$$x_{i_{MAX}} = 100 \quad \text{ou} \quad y_i = \frac{x_i}{100}$$

tem-se

$$\text{MIN } Z = -21y_1 - 12y_2$$
$$f_1 = y_1 + 2y_2 - 0{,}31 \leq 0$$
$$f_2 = y_1 + y_2 - 0{,}11 \leq 0$$
$$f_3 = y_2 - 0{,}12 \leq 0$$
$$\phi_1 = -y_1 \leq 0$$
$$\phi_2 = -y_2 \leq 0.$$

b.3) O sistema gradiente terá a forma

$$-\frac{dy_1}{dt} = -21 + \alpha\left[f_1\ \delta f_1 + f_2\ \delta f_2 + y_1\ \delta(-y_1)\right] .$$

$$-\frac{dy_1}{dt} = -12 + \alpha\left[f_2\ \delta f_2 + f_3\ \delta f_3 + y_2\ \delta(-y_2)\right] .$$

b.4) A Figura 5.2 mostra o circuito com diodos utilizados na solução analógica.

b.5) O quadro apresenta os resultados obtidos para as diversas condições iniciais introduzidas e os diversos valores de α.

α	CONDIÇÕES INICIAIS		RESULTADOS		TEMPO DE OBTENÇÃO DO RESULTADO(S)	
	x_1	x_2	x_1	x_2	x_1	x_2
1	0	0	0,4100	-0,8989	21	25
1	-0,4	-0,4	0,4102	-0,9005	22	27
1	-0,6	-0,6	0,4102	-0,9005	27	33
10	0	0	1,405	-0,0899	4	6
10	-0,7	0	1,407	-0,0920	3	4
10	0,4	0,4	1,404	-0,0900	3	4
20	0	0	1,256	-0,0458	2,5	4
20	0,4	0,4	1,256	-0,0458	2	2
20	0,7	0	1,256	-0,0470	1,5	1,5
100	0	0	11,357	-0,0101	2	2
100	0,1	0,1	11,343	-0,0080	2,5	2,5
100	0	0	11,343	-0,0080	15	(1)17
200	0	0	11,105	-0,0042	1,5	1,5
200	0	0	11,095	-0,0042	15	(1)15
1 000	0	0	11,072	-0,0007	29	(1)29

A melhor solução obtida no computador analógico foi para um valor máximo de $\alpha = 1.000$ e

$x_1 = 11,072$

$x_2 = -0,0007$

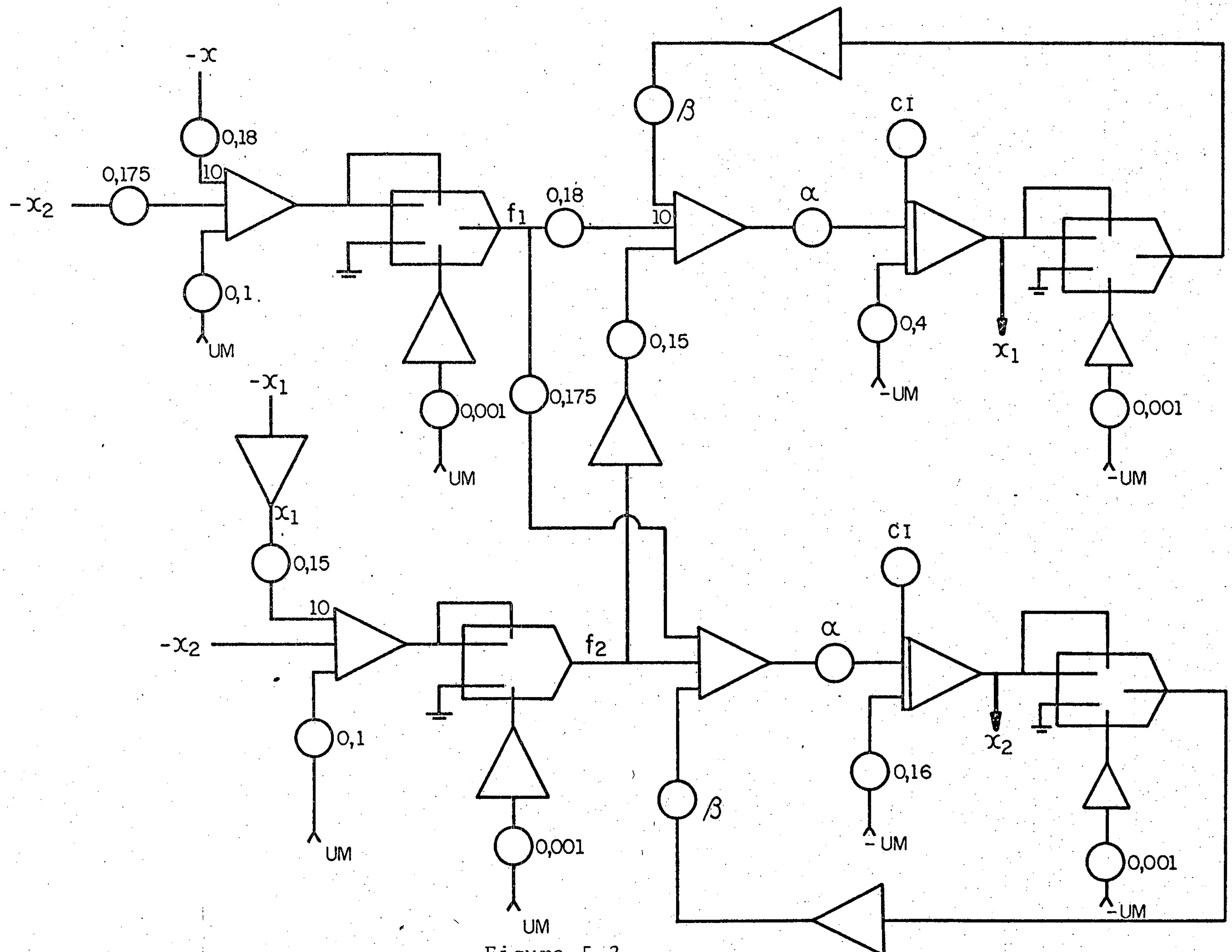
MIN Z = -232,5

quando a solução ótima é

$x_1 = 11$

$x_2 = 0$

MIN Z = -231.



-x
0,18
10
0,175
$-x_2$
0,1
UM
0,001
UM
f_1
0,18
10
0,175
0,15
β
α
CI
0,4
-UM
x_1
0,001
-UM
$-x_1$
x_1
0,15
10
$-x_2$
0,1
UM
0,001
UM
f_2
β
α
CI
0,16
-UM
x_2
0,001
-UM

Figura 5.2.

c) O problema, resolvido da mesma forma escalonada no computador IBM 1130, utilizando o programa LPMOSS[13], apresentou o seguinte resultado:

$x_1 = 11,0$

$x_2 = 0,0$

$\text{MIN } Z = 231,0$

tendo-se que o tempo de busca dos resultados foi de 10 minutos.

d) Comentários.

O problema foi resolvido escalonado no computador digital e, como pode ser visto, a precisão é comparável com a do computador analógico.

A solução analógica utilizou os circuitos com diodos e estão indicados no diagrama da figura através do bloco de sub-rotina com o nome de $\delta f(\underline{X})$.

O tempo foi medido fora dos computadores através de cronômetro; portanto, é bastante impreciso; está citado aqui apenas para dar uma idéia de grandeza, não servindo para uma análise mais apurada.

As soluções assinaladas com (1) no quadro, foram obtidas introduzindo-se uma redução no valor de c_j, isto é, utilizando-se a função

$$W(\underline{X},\alpha,\mu) = \mu Z(\underline{X}) + \alpha U(\underline{X})$$

onde $\mu = 0,1$. Como era de se esperar, isto aumenta o tempo para o sistema atingir o estado estacionário.

CAPÍTULO VI

CONCLUSÕES

A prova do método de Rybashov apresentada no Anexo 1 é quase intuitiva. Procurou-se exemplificar este método com a realização de problemas no computador analógico-híbrido.

Como não foram resolvidos outros problemas além daqueles de programação linear, não é possível verificar o método de maneira genérica. Mas se pressupõe que o mesmo é aplicável na solução de problemas de programação não-linear mas convexa.

As soluções obtidas usando o método analógico coincidiram com as soluções obtidas usando o método digital e, obviamente, com as soluções esperadas, ainda que os problemas solucionados sejam simples e acadêmicos.

No que se refere a tempo, apesar da solução analógica ser obtida muito mais rapidamente, ela apresenta o inconveniente de demandar um longo tempo de programação (escalonamento e montagem).

No método analógico, não é preciso encontrar uma solução inicial (solução básica possível), pois ficou comprovado que o problema converge qualquer que seja a condição inicial, isto é, a solução de minimizar $W(\underline{X},\alpha)$ coincide com o problema de programação linear.

Esta coincidência de valores é realmente função de α. Dependendo do problema, as condições iniciais em nada influenciam o tempo de convergência para a solução estável.Como o processo analógico não é iterativo, não ocorre degeneração como definida anteriormente. Não foi possível testar o "cycling" devido ao fato de que nenhum dos computadores utilizados comportava um problema tão grande tal que ocorresse o "cycling". Em qualquer caso, é necessário usar decomposição para problemas grandes.

O tamanho do problema é o que acarreta a maior desvantagem do computador analógico, na solução de problemas de programação linear, pois envolve um grande número de variáveis e restrições.

Outra desvantagem é que, em programação linear, são realizadas inúmeras operações puramente algébricas o que gera instabilidade em todo o sistema e, como o número de variáveis é grande, o ajuste da situação estável é trabalhosa. Entretanto, em programação não-linear, é provavelmente mais eficiente e preciso usar o computador analógico pela utilização fácil de blocos não-lineares.

Quando o número de variáveis e restrições é pequeno,o computador analógico apresenta a vantagem de poder variar os parâmetros do problema através da variação de potenciômetros sem a necessidade da reprodução de todo o problema, permitindo, assim, uma análise de sensibilidade.

Quanto aos circuitos realizados:

- aqueles a diodo externo possuem, como se sabe, uma baixa precisão devido à sua característica não-linear quando próximos de zero. Como se procuram pontos que transformem as restrições em igualdade, isto pode ser prejudicial;

- nos circuitos a comparadores eletrônicos, é necessário acrescentar um valor diferente de zero para que ele opere corretamente. Isto também constitui uma imprecisão.

Sugere-se, como trabalho futuro, um estudo generalizado de problemas de programação matemática segundo as formas apresentadas por Arrow e Hurwicz[3] ou por G.Hadley[10] que já desenvolveram o chamado método diferencial do gradiente para resolver problemas de maximizar uma função côncava $f(\underline{X})$ sobre um conjunto de restrições

$$g_i(\underline{X}) \leq 0 \qquad (i=1,\ldots m)$$

onde $\underline{X} > 0$ e $g_i(\underline{X})$ são funções convexas. O método é dito diferencial, porque o passo tomado em cada iteração é muito pequeno, o que exige um grande número de iterações.

A idéia fundamental é, partindo da função de Lagrange

$$L(\underline{X}, \underline{\lambda}) = f(\underline{X}) + \sum_{i=1}^{m} \lambda_i \left[b_i + g_i(\underline{X}) \right]$$

resolver o sistema de equações diferenciais

$$\frac{dx_j}{dt} = 0 \quad \text{se} \quad \frac{\partial L}{\partial x_j} < 0 \quad \text{e} \quad x_j = 0$$

$$\frac{dx_j}{dt} = \frac{\partial F}{\partial x_j} \qquad \text{nos outros pontos}$$

e

$$\frac{d\lambda_i}{dt} = 0 \quad \text{se} \quad \frac{\partial L}{\partial \lambda_i} > 0 \quad \text{e} \quad \lambda_i = 0$$

$$\frac{d\lambda_i}{dt} = -\frac{\partial L}{\partial \lambda_i} \quad \text{nos outros pontos.}$$

Sugere-se estudar, detalhadamente, a solução de problemas de programação não-linear, explorando as potencialidades que o computador analógico apresenta nesta área.

Sugere-se, também, um estudo do problema de otimização através de um sistema híbrido, utilizando-se as vantagens de rápida solução de equações diferenciais desta máquina segundo é indicado nos artigos de J.A.Gibson[8] e R.Gonzalez[9].

ANEXO I

PROVA DO TEOREMA 4.1

Considerem-se, em torno do ponto $\underline{X}^o$, duas esferas de raio δ_1 e δ respectivamente, onde $\delta > \delta_1$ de acordo com a Figura A.1.

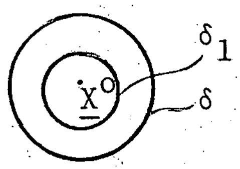


Figura A.1

Considere-se o conjunto M dos pontos pertencentes à esfera de raio δ e não-pertencentes à esfera de raio δ_1 (incluindo os pontos das superfícies esféricas) tais que

$$F(\underline{X}) \leq F(\underline{X}^o) \qquad \underline{X} \in M.$$

Se o conjunto M é vazio, então fora de δ_1, mas em δ, $F(\underline{X}) > F(\underline{X}^o)$ e

$$W(\underline{X},\alpha) = F(\underline{X}) + \alpha U(\underline{X}) > F(\underline{X}^o)$$

enquanto que $W(\underline{X}^o,\alpha) = F(\underline{X}^o)$. Dessa forma, quando M é um conjunto vazio, para qualquer α, o ponto de mínimo de $W(\underline{X},\alpha)$ pertence à esfera δ_1. Se M não é um conjunto vazio, é possível encontrar um limite inferior exato

$$g = \inf U(\underline{X}) \qquad (\underline{X}) \in M)$$

que é positivo. Entretanto, $J(\underline{X})$ é nulo somente em R, e o conjunto M não pode ter ponto em comum com R, uma vez que em R se tem a restrição $F(\underline{X}) > F(\underline{X}^o)$.

Fazendo

$$\alpha = \frac{F(\underline{X}^o) - \overline{F}}{g}$$

onde

$$\bar{F} = \inf F(\underline{X}), \; \underline{X} \in M$$

pode-se escrever

$$W(\underline{X},\alpha) = F(\underline{X}) + \alpha.U(\underline{X}) \geq \inf F(\underline{X}) + \alpha \inf U(\underline{X})$$

$$W(\underline{X},\alpha) = \bar{F} + \alpha.g > F(\underline{X}^o).$$

Então, para α definido acima,

$$W(\underline{X},\alpha) > F(\underline{X}^o) \qquad (\underline{X} \in M)$$

e também fora de M, $W(\underline{X},\alpha) > F(\underline{X}^o)$ por definição

Assim, a função convexa $W(\underline{X},\alpha)$ tem valor maior em pontos da esfera δ do que em um ponto da esfera δ_1. Conclui-se, então, que no interior da esfera δ_1 existe um mínimo de $W(\underline{X},\alpha)$ e, consequentemente, em virtude da convexidade de $W(\underline{X},\alpha)$, não existe mínimo fora de δ_1.

Dessa forma, o conjunto de pontos π no qual a função atinge um mínimo absoluto é limitado, fechado e depende de α:

$$\pi = \pi\ (\alpha).$$

Em virtude da arbitrariedade do número δ_1, ele pode ser tal que

$$\mathrm{MAX}_i \ \{|x_i - x_1^o|\} \leq \gamma \qquad \text{A.1}$$

Isso prova que existe um mínimo da função $W(\underline{X},\alpha)$ em um conjunto limitado, fechado $\pi(\alpha)$.

Por definição, qualquer $\underline{X}$,

$$W(\bar{\underline{X}},\alpha) \leq W(\underline{X},\alpha) \qquad \bar{\underline{X}} \in \pi(\alpha),$$

em particular para $\underline{X} = \underline{X}^o$

$$W(\bar{\underline{X}},\alpha) \leq W(\underline{X}^o,\alpha) = F(\underline{X}^o) + \alpha U(\underline{X}^o) = F(\underline{X}^o).$$

Ao mesmo tempo, uma vez que $\alpha > 0$ e $U(\underline{X}) \geq 0$,

$$W(\underline{\overline{X}},\alpha) = F(\underline{\overline{X}}) + \alpha U(\underline{\overline{X}}) \geq F(\underline{\overline{X}}),$$

então, para qualquer $\underline{\overline{X}} \in \pi(\alpha)$, é válido o seguinte:

$$F(\underline{\overline{X}}) \leq W(\underline{\overline{X}},\alpha) \leq F(\underline{X}^o).$$

Mas, quando $\alpha \to \infty$, o conjunto $\pi(\alpha)$ se torna no ponto $\underline{X}^o$ e $F(\underline{\overline{X}}) \to F(\underline{X}^o)$, isto é,

$$W(\underline{\overline{X}},\alpha) \to F(\underline{X}^o) \qquad \alpha \to \infty$$

Por continuidade, encontra-se um δ_1 tal que

$$|W(\underline{\overline{X}},\alpha) - F(\underline{X}^o)| \xi \qquad \text{A.2}$$

seja satisfeita.

Escolhendo um δ_1 tal que seja o menor determinado pelas restrições A.1 e A.2, conclui-se a veracidade do teorema.

ANEXO II

SOLUÇÃO LPOMSS DOS PROBLEMAS-EXEMPLO

EXEMPLO 1

```
// JOB
// XEQ MOSS

INPUT
NAME            PROB9
ENDATA
PROBLEM'PROB9    ' CONTAINS
     3 ROWS
     0 SELECTED ROWS
     5 VARIABLES
     0 SELECTED COLUMNS
     1 BOUNDS
     0 RHS'S
     0 RANGES
     6 COLUMN ELEMENTS
     1 LOWER BOUND ELEMENTS
     3 UPPER BOUND ELEMENTS
     0 RHS ELEMENTS
     0 RANGE ELEMENTS
 MOVE
     DATA       PROB9
     MINIMIZE   FOB9
     BOUNDS     LIM1
 ENDATA
 OPTIMIZE
 ITERATION INFEASIBILITY  VALUE OF
  NUMBER        COUNT      'FOB9     '

      0             0          0.
 SOLUTION OPTIMUM
 LPSOLUTION
```

VARIABLE	TYPE	ENTRIES	SOLUTION ACTIVITY	UPPER BOUND	LOWER BOUND	CURRENT COST	REDUCED COST
X1	LL	3	0.000	********	0.000	0.400	-0.400
FOB9	B*	0	0.000	********	******	-1.000	1.000
X2	LL	3	0.000	********	0.000	1.600	-1.800
E19	B*	0	0.000	1.000	0.000	0.000	0.000
E29	B*	0	0.000	1.000	0.000	0.000	0.000

EXEMPLO 2

```
INPUT
NAME              PROB80
ENDATA
PROBLEM 'PROB80    '   CONTAINS
    4 ROWS
    0 SELECTED ROWS
    6 VARIABLES
    0 SELECTED COLUMNS
    1 BOUNDS
    0 RHS'S
    0 RANGES
    7 COLUMN ELEMENTS
    1 LOWER BOUND ELEMENTS
    4 UPPER BOUND ELEMENTS
    0 RHS ELEMENTS
    0 RANGE ELEMENTS
MOVE
    DATA          PROB80
    MINIMIZE      FOB80
    BOUNDS        LIM1
ENDATA
OPTIMIZE
ITERATION  INFEASIBILITY  VALUE OF
 NUMBER        COUNT      'FOB80    '

    0              0             0.000
    1              0            -0.231
SOLUTION OPTIMUM
LPSOLUTION
```

VARIABLE	TYPE	ENTRIES	SOLUTION ACTIVITY	UPPER BOUND	LOWER BOUND	CURRENT COST	REDUCED COST
X1	B*	3	0.110	*********	0.000	-2.100	0.000
FOB80	B*	0	-0.231	*********	*******	-1.000	1.000
X2	LL	4	0.000	*********	0.000	-1.200	-0.900
E180	B*	0	0.110	0.310	0.000	0.000	0.000
E280	UL	0	0.110	0.110	0.000	0.000	-2.100
E380	B*	0	0.000	0.120	0.000	0.000	0.000

NOTAÇÃO E SIMBOLOGIA

a_{ij}, b_i, c_j, x_i, y_i ...: escalares

$\underline{A}$, $\underline{B}$, $\underline{X}$, $\underline{c}$, $\underline{b}$ ...: matrizes e vetores

$\underline{c}^T$: transposta da matriz ou vetor

$\underline{B}^{-1}$: matriz inversa

I : matriz identidade

J : matriz Jacobiana

|J|: determinante da matriz Jacobiana

E_n : espaço euclidiano

{.} : conjunto

$\in$: pertence

$\cap$: intersecção

|.| : módulo

$\forall$: para todo

| : tal que

>> : muito maior

$\rightarrow$: tende à

BIBLIOGRAFIA

1 ABLOW, C.M. e BRIGHAM, G. *An analog solution of programming problems*. OPER.RESEARCH, USA, v. 3, 1955, p. 388-94.

2 ACKOFF, Russel L. *Progress in operations research*. USA, John Wiley, 1961.

3 ARROW, K.; HURWICZ, L.; UZAWA, H. *Studies in linear and non linear programming*. USA, Stanford University Press, 1972.

4 DANTZIG, G. *Linear programming and extensions*. USA, Princeton University Press, 1963.

5 DUDNIKOV, E.E. and RYBASHOV, M.V. *Methods for the solution of mathematical programming problems on general purpose analog computers (survey)*. Avtomatika i Telemekhanika, URSS, n.5, 1967. p. 109-52.

6 GARVIN, W. *Introduction to linear programming*. USA, Mc Graw Hill, 1960.

7 GASS, S.I. *Linear programming - methods and applications*. USA, Mc Graw Hill Book Company, 1964.

8 GIBSON, J.A. and MARKS, T.W. *A unifying zero-zone* Function Approach to restrained System Optimization. USA, IEEE Transc. SMC, March, 1973. p. 187-93.

9 GONZALEZ, R. *An optimization study on a hybrid computer*. Annales de l'AICA, n.3, 1970. p. 138-48.

10 HADLEY, G. *Non linear and dynamic programming*. USA, Addison - Wesley, 1964.

11 HAUSNER, A. *Analog and analog-hybrid computer programming*. Prentice Hall, USA, 1971.

12 HU, T.C. *Integer programming and net work flows*. USA, Addison-Wesley, 1970.

13 IBM-1130 *Linear programming* - Mathematical Optimization Subroutine System (program reference manual).

14 KARPINSKAYA, I.N. and RIBASHOV, M.V. *A method for solving the linear programming problem with an electronic model.* Avtomatika i Telemekhanika, URSS, v.24, n.3, 1963. p. 361-8.

15 KARPINSKAYA, I.N. *Method of penalty function and the foundations of Pyne's method.* Avtomatika i Telemekhanika, URSS, v. 2, n.1, 1967. p.140-6.

16 KORN, G. and KORN, T. *Electronic analog and hybrid computers.* USA, Mc Graw Hill, 1972.

17 LONGLEY, D. *An analogue computer application in operations research.* Electronic Engineering, England, june, 1964, p. 378-81.

18 NEELY, P.L. and ROBERTS, A.P. *Am improvised analogue technique for solving trajectory optimization problems.* INT. JOURNAL OF CONTROL, 1971, v.13, n.1. p. 33-52.

19 PYNE, I.B. *Linear programming on an electronic analog computer.* AIEE TRANS, USA, may 1966, p.139-43.

20 ROGERS, A.E. and CONNOLLY, T.W. *Analog computation in engineering design.* USA, Mc Graw Hill, 1960.

21 RYBASHOV, M.V. *Analog solution of algebraic and transcendental equations by gradient method.* Avtomatika i Telemekhanika, URSS, v.22, 1961. p. 77-88.

22 _____. *The gradient method for solving convex programming on electronic analog computers.* Avtomatika i Telemekhanika, URSS, v.26, n.11, nov. 1965. p. 1955-67.

23 _____. *The gradient method of solving linear and quadratic programming problems on electronic analog computers.* Avtomatika i Telemekhanika, URSS, v.26, n.12, dec.1965. p.2151-62.

24 _____. *Circuits for analog computers solution of systems of linear equalities, inequalities, linear programming problem and matrix games.* Avtomatika i Telemekhanika, URSS, march 1969, n.3. p. 141-64.

25 _____. *Insensitivity of gradient systems in the solution of linear problems on analog computers.* Avtomatika i Telemekhanika, URSS, n.10, oct. 1969. p.147-55.

26 SAATY, T. *Mathematical methods of operations research.* USA, Mc Graw Hill, 1959.

27 SAKSENA, C.P. and COLE, A.J. *The bounding hyperplane method of linear programming.* OPER. RESEARCH, USA, jan. 1971.p.1-18.

28 SANTOS, José Abel Royo dos. *Computadores analógicos-programação avançada.* EFEI, Itajubá, Brasil, 1970.

29 WILKINS, B.R. *Analogue and iterative methods in computation, simulation and control.* CHAPMAN AN HALL, England, 1970.